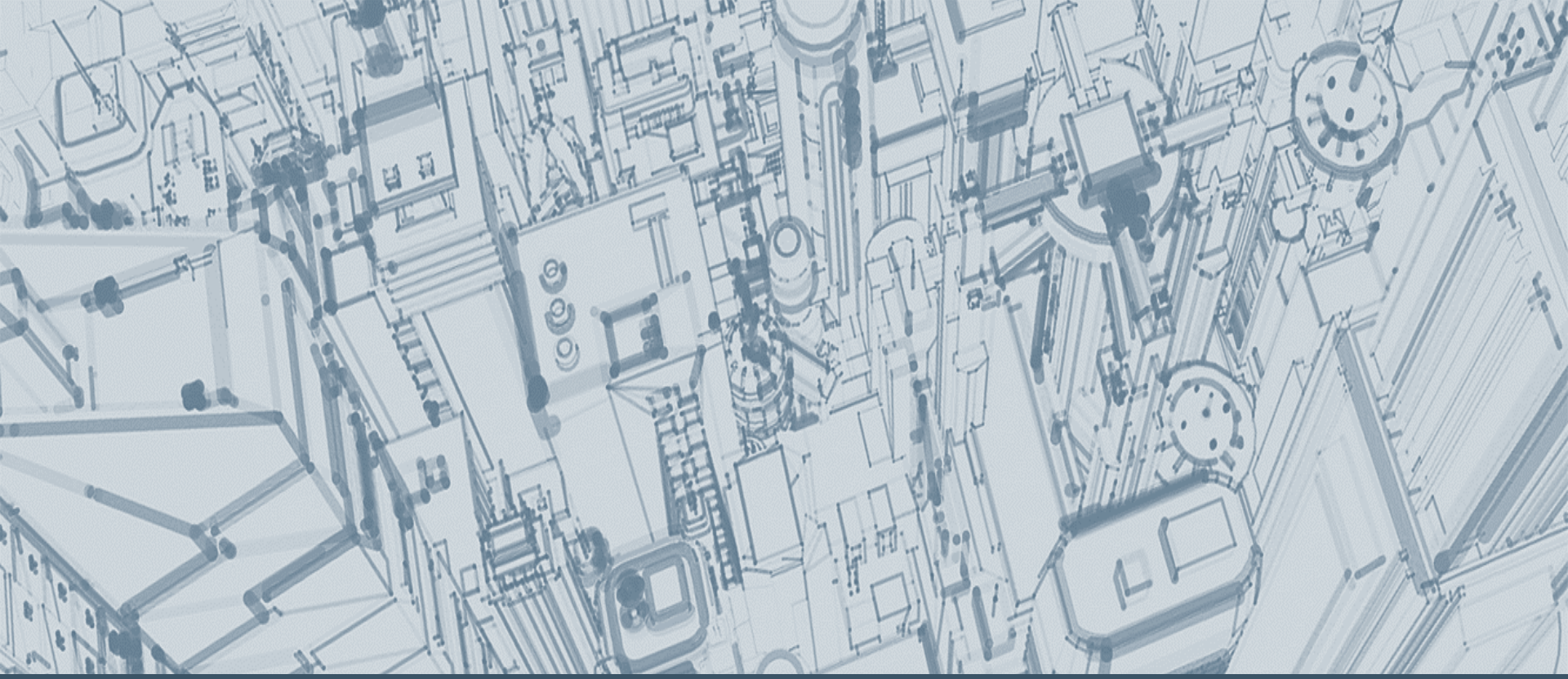


LabA UG871: Design Optimization

Speaker: Hua-Yang Weng Date: 2021.10.17

https://github.com/Yuoto/ug871_design_optimization

Take home Idea

- Pipeline @ Different Levels
  - Loop level
    - Product, col, row
  - Function level

- Appendix
  - Reshape v.s. Partition
  - Streaming & bugs

Matrix Multiplication

- $O(n^3)$: Three loops

```
#include "matrixmul.h"

void matrixmul(
      mat_a_t a[MAT_A_ROWS][MAT_A_COLS],
      mat_b_t b[MAT_B_ROWS][MAT_B_COLS],
      result_t res[MAT_A_ROWS][MAT_B_COLS])
{
  int tmp = 0;
  // Iterate over the rows of the A matrix
  Row: for(int i = 0; i < MAT_A_ROWS; i++) {
    // Iterate over the columns of the B matrix
    Col: for(int j = 0; j < MAT_B_COLS; j++) {
      res[i][j] = 0;
      // Do the inner product of a row of A and col of B
      Product: for(int k = 0; k < MAT_B_ROWS; k++) {
        res[i][j] += a[i][k] * b[k][j];
      }
    }
  }
}
```

```
#ifndef __MATRIXMUL_H__
#define __MATRIXMUL_H__

#include <cmath>
using namespace std;

// Compare TB vs HW C-model and/or RTL
#define HW_COSIM

#define MAT_A_ROWS 3
#define MAT_A_COLS 3
#define MAT_B_ROWS 3
#define MAT_B_COLS 3

typedef char mat_a_t;
typedef char mat_b_t;
typedef short result_t;
typedef char in_mat_t;

// Prototype of top level function for C-synthesis
void matrixmul(
      mat_a_t a[MAT_A_ROWS][MAT_A_COLS],
      mat_b_t b[MAT_B_ROWS][MAT_B_COLS],
      result_t res[MAT_A_ROWS][MAT_B_COLS]);

#endif // __MATRIXMUL_H__ not defined
```

No pragmas

Vivado_hls

- No pipeline, loop enter/exit causes extra delays.

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	13.33	3.576	1.67

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
79	79	79	79	none

Detail

Instance

Loop

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row	78	78	26	-	-	3	no
+ Col	24	24	8	-	-	3	no
++ Product	6	6	2	-	-	3	no

Vitis_hls

- Default pipelined @ col loop (flatten also)
- Still II=2 need to be fixed manually.

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00 ns	3.372 ns	1.35 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
29	29	0.145 us	0.145 us	30	30	none

Detail

Instance

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row_Col	27	27	12	2	1	9	yes

Pipeline @ Different Levels

Loop level: Product loop

Vivado_hls

- Dependency about res[i][j] for different k iteration ◦ **Res is simultaneously reading and writing → II=2**

```
res[i][j] += a[i][k]*b[k][j]
```

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	13.33	4.306	1.67

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
82	82	82	82	none

Detail

Instance

Loop

```
// Iterate over the rows of the A matrix
Row: for(int i = 0; i < MAT_A_ROWS; i++) {
    // Iterate over the columns of the B matrix
    Col: for(int j = 0; j < MAT_B_COLS; j++) {
        res[i][j] = 0;
        // Do the inner product of a row of A and col of B
        Product: for(int k = 0; k < MAT_B_ROWS; k++) {
            res[i][j] += a[i][k] * b[k][j];
```

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row_Col	81	81	9	-	-	9	no
+ Product	6	6	2	2	1	3	yes

Vitis_hls

- Product loop is also flattened
- Hence, trip count = 27
- Same II = 2 dependency problem

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00 ns	3.551 ns	1.35 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
62	62	0.310 us	0.310 us	63	63	none

Detail

Instance

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row_Col_Product	60	60	9	2	1	27	yes

Loop level: Col loop (1/5)

- All loops in the hierarchy below are automatically unrolled.
- Unrolling Product loop requires more ports during k iteration
- Dual port a, b memory port is not enough, hence II=2 for

```
INFO: [XFORM 203-502] Unrolling all sub-loops inside loop 'Col' (matrixmul.cpp:56) in
function 'matrixmul' for pipelining.
INFO: [XFORM 203-501] Unrolling loop 'Product' (matrixmul.cpp:59) in function
'matrixmul' completely.
INFO: [XFORM 203-541] Flattening a loop nest 'Row' (matrixmul.cpp:54:37) in function
'matrixmul'.
...
...
INFO: [SCHED 204-61] Pipelining loop 'Row_Col'.
WARNING: [SCHED 204-69] Unable to schedule 'load' operation ('a_load_1',
matrixmul.cpp:60) on array 'a' due to limited memory ports.
```

```
// Iterate over the rows of the A matrix
Row: for(int i = 0; i < MAT_A_ROWS; i++) {
  // Iterate over the columns of the B matrix
  Col: for(int j = 0; j < MAT_B_COLS; j++) {
    res[i][j] = 0;
    // Do the inner product of a row of A and col of B
    Product: for(int k = 0; k < MAT_B_ROWS; k++) {
      res[i][j] += a[i][k] * b[k][j];
```

Loop level: Col loop (2/5)

- Array Reshape (Vivado_hls)

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	matrixmul	return value
ap_rst	in	1	ap_ctrl_hs	matrixmul	return value
ap_start	in	1	ap_ctrl_hs	matrixmul	return value
ap_done	out	1	ap_ctrl_hs	matrixmul	return value
ap_idle	out	1	ap_ctrl_hs	matrixmul	return value
ap_ready	out	1	ap_ctrl_hs	matrixmul	return value
a_address0	out	2	ap_memory	a	array
a_ce0	out	1	ap_memory	a	array
a_q0	in	32	ap_memory	a	array
b_address0	out	2	ap_memory	b	array
b_ce0	out	1	ap_memory	b	array
b_q0	in	32	ap_memory	b	array
res_address0	out	4	ap_memory	res	array
res_ce0	out	1	ap_memory	res	array
res_we0	out	1	ap_memory	res	array
res_d0	out	16	ap_memory	res	array

Performance Estimates

```
HLS ARRAY_RESHAPE reshape variable=a complete dim=2
HLS ARRAY_RESHAPE reshape variable=b complete dim=1
```

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	13.33	7.566	1.67

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
11	11	11	11	none

Detail

Instance

Loop

	Latency			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row_Col	9	9	2	1	1	9	yes

```
// Iterate over the rows of the A matrix
Row: for(int i = 0; i < MAT_A_ROWS; i++) {
  // Iterate over the columns of the B matrix
  Col: for(int j = 0; j < MAT_B_COLS; j++) {
    res[i][j] = 0;
    // Do the inner product of a row of A and col of B
    Product: for(int k = 0; k < MAT_B_ROWS; k++) {
      res[i][j] += a[i][k] * b[k][j];
```

Loop level: Col loop (3/5)

- Array Reshape (Vitis_hls) II=1
- However, with pipeline depth = 7，latency 16 cycles, 80ns

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00 ns	3.570 ns	1.35 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
16	16	80.000 ns	80.000 ns	17	17	none

Detail

Instance

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row_Col	14	14	7	1	1	9	yes

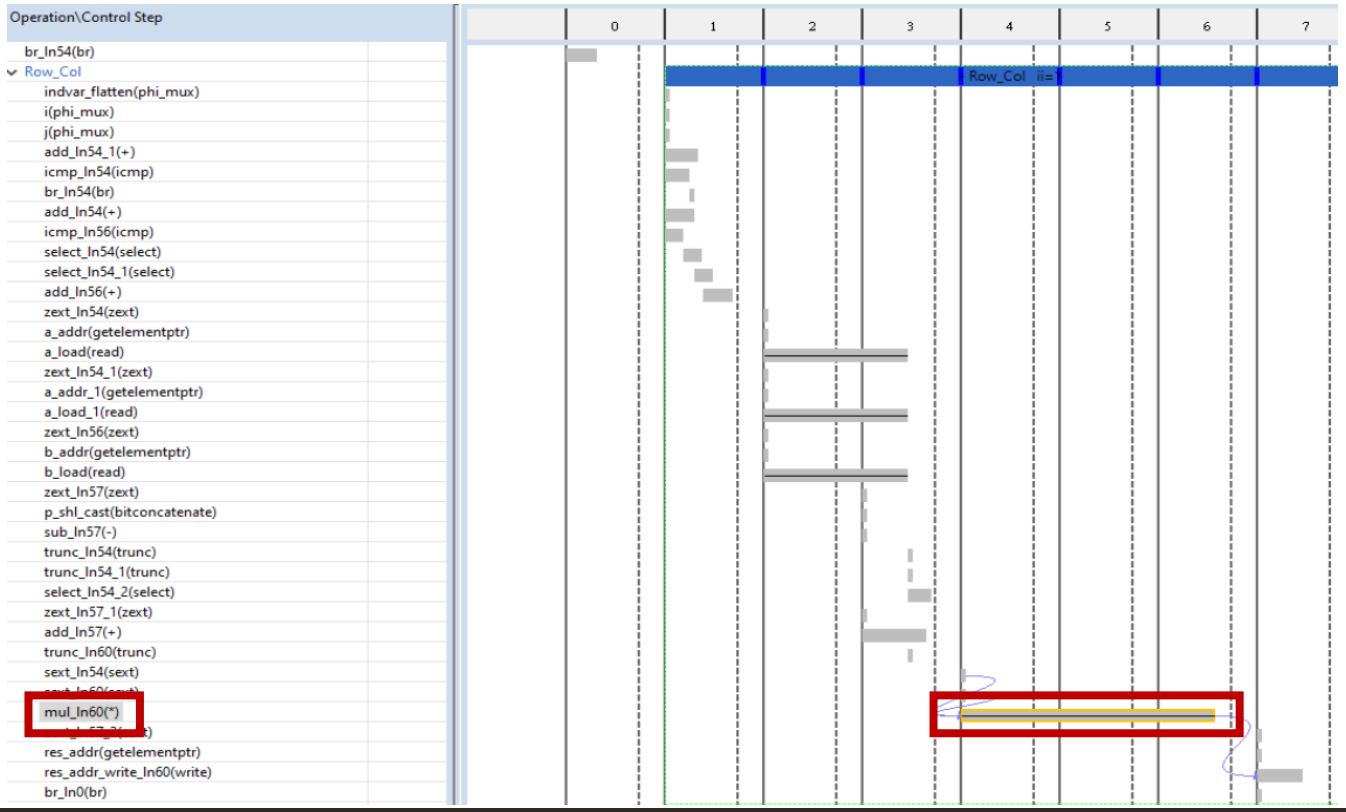
Utilization Estimates

Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	80	-
FIFO	-	-	-	-	-
Instance	-	1	0	0	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	88	-
Register	-	-	195	64	-
Total	0	1	195	232	0
Available	280	220	106400	53200	0
Utilization (%)	0	~0	~0	~0	0

Loop level: Col loop (4/5)

- There is only one multiplier & takes 3 cycles



Loop level: Col loop (5/5)

- 200MHz, pipeline depth = 7
- 75MHz，pipeline depth = 2，　less resource, still 1 Multiplier
- However, latency = 160ns　→ 2x more than previous case

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	13.33 ns	10.062 ns	1.67 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
12	12	0.160 us	0.160 us	13	13	none

Detail

Instance

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row_Col	10	10	2	1	1	9	yes

200MHz

Utilization Estimates

Summary

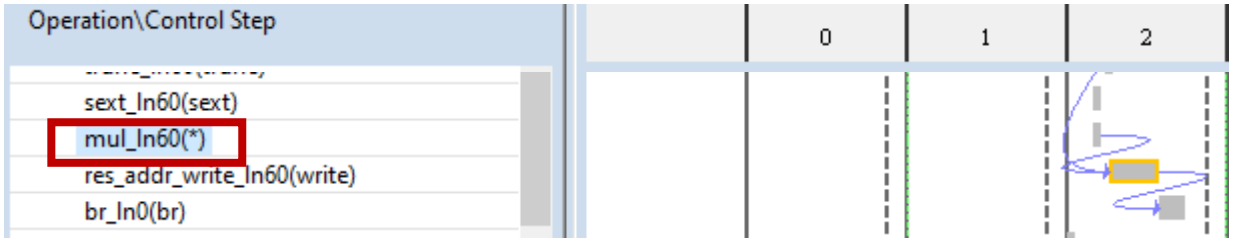
Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	80	-
FIFO	-	-	-	-	-
Instance	-	1	0	0	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	88	-
Register	-	-	[illegible]	[illegible]	-
Total	0	1	195	232	0
Available	280	220	106400	53200	0
Utilization (%)	0	~0	~0	~0	0

75MHz

Utilization Estimates

Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	78	-
FIFO	-	-	-	-	-
Instance	-	0	0	41	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	65	-
Register	-	-	19	-	-
Total	0	0	19	184	0
Available	280	220	106400	53200	0
Utilization (%)	0	0	~0	~0	0



Loop level: Row loop (1/3)

```
// Iterate over the rows of the A matrix
Row: for(int i = 0; i < MAT_A_ROWS; i++) {
    // Iterate over the columns of the B matrix
    Col: for(int j = 0; j < MAT_B_COLS; j++) {
        res[i][j] = 0;
        // Do the inner product of a row of A and col of B
        Product: for(int k = 0; k < MAT_B_ROWS; k++) {
            res[i][j] += a[i][k] * b[k][j];
```

- Array Reshape (Vitis_hls) II=1
- Pipeline depth = 18，latency 105ns

```
matrixmul
  % HLS TOP name=matrixmul
  a
  % HLS ARRAY_RESHAPE variable=a complete dim=1
  % HLS ARRAY_RESHAPE variable=a complete dim=2
  b
  % HLS ARRAY_RESHAPE variable=b complete dim=2
  % HLS ARRAY_RESHAPE variable=b complete dim=1
  res
  % HLS ARRAY_RESHAPE variable=res complete dim=2
  Row
    % HLS PIPELINE
    Col
      Product
```

Worse than col pipeline

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00 ns	3.853 ns	1.35 ns

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
21	21	0.105 us	0.105 us	22	22	none

Detail

Instance

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row	19	19	18	1	1	3	yes

Utilization Estimates

Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	6	-	-	-
Expression	-	-	1250	1536	-
FIFO	-	-	-	-	-
Instance	-	3	2046	1791	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	65	-
Register	-	-	958	160	-
Total	0	9	4254	3552	0
Available	280	220	106400	53200	0
Utilization (%)	0	4	3	6	0

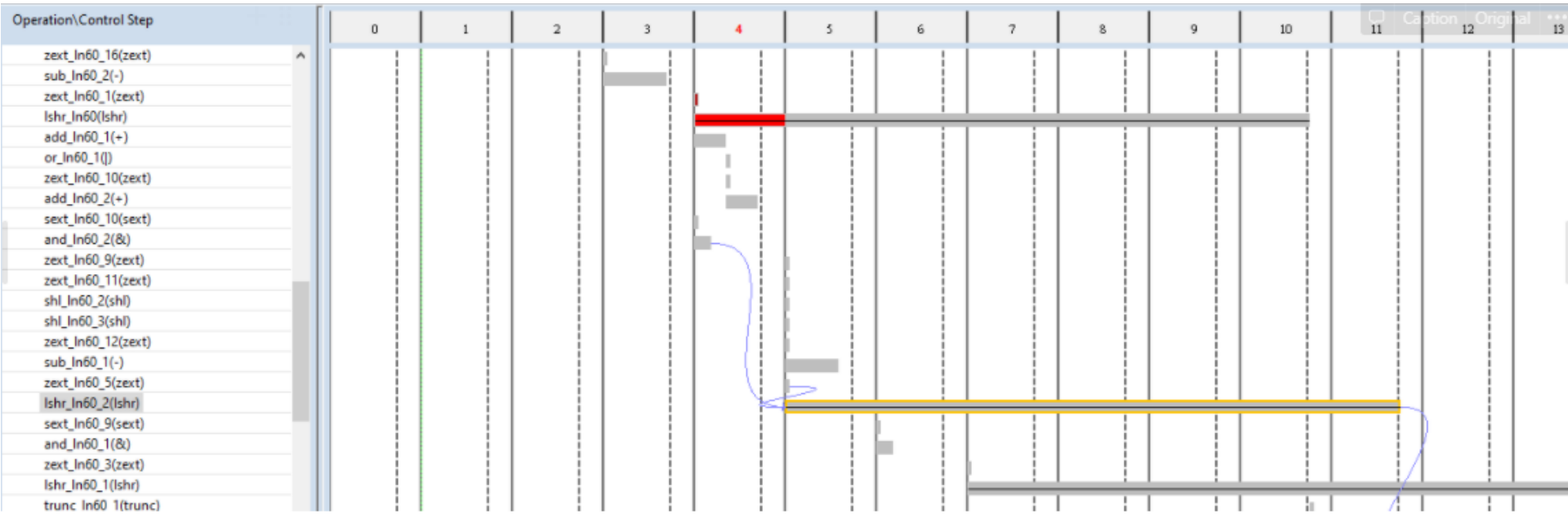
Loop level: Row loop (2/3)

- Array Reshape (Vitis_hls) II=1
  - Very long pipeline for shifter (Not always)
  - 9 Multipliers
  - This is resolved in array partition

Interface

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	matrixmul	return value
ap_rst	in	1	ap_ctrl_hs	matrixmul	return value
ap_start	in	1	ap_ctrl_hs	matrixmul	return value
ap_done	out	1	ap_ctrl_hs	matrixmul	return value
ap_idle	out	1	ap_ctrl_hs	matrixmul	return value
ap_ready	out	1	ap_ctrl_hs	matrixmul	return value
a	in	72	ap_none	a	pointer
b	in	72	ap_none	b	pointer
res_address0	out	2	ap_memory	res	array
res_ce0	out	1	ap_memory	res	array
res_we0	out	8	ap_memory	res	array
res_d0	out	64	ap_memory	res	array



Loop level: Row loop (3/3)

- Array Partition (Vitis_hls) II=1
- Pipeline depth = 9，latency 60ns
- 9 multipliers

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00 ns	2.936 ns	1.35 ns

Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	6	-	-	-
Expression	-	-	0	86	-
FIFO	-	-	-	-	-
Instance	-	3	0	0	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	56	-
Register	-	-	571	128	-
Total	0	9	571	270	0
Available	280	220	106400	53200	0
Utilization (%)	0	4	~0	~0	0

Latency

Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
12	12	60.000 ns	60.000 ns	13	13	none

Detail

Instance

Loop

	Latency (cycles)			Initiation Interval			
Loop Name	min	max	Iteration Latency	achieved	target	Trip Count	Pipelined
- Row	10	10	9	1	1	3	yes

Summary

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	matrixmul	return value
ap_rst	in	1	ap_ctrl_hs	matrixmul	return value
ap_start	in	1	ap_ctrl_hs	matrixmul	return value
ap_done	out	1	ap_ctrl_hs	matrixmul	return value
ap_idle	out	1	ap_ctrl_hs	matrixmul	return value
ap_ready	out	1	ap_ctrl_hs	matrixmul	return value
a_0_0	in	8	ap_none	a_0_0	pointer
a_0_1	in	8	ap_none	a_0_1	pointer
a_0_2	in	8	ap_none	a_0_2	pointer
a_1_0	in	8	ap_none	a_1_0	pointer
a_1_1	in	8	ap_none	a_1_1	pointer
a_1_2	in	8	ap_none	a_1_2	pointer
a_2_0	in	8	ap_none	a_2_0	pointer
a_2_1	in	8	ap_none	a_2_1	pointer
a_2_2	in	8	ap_none	a_2_2	pointer
b_0_0	in	8	ap_none	b_0_0	pointer
b_0_1	in	8	ap_none	b_0_1	pointer
b_0_2	in	8	ap_none	b_0_2	pointer
b_1_0	in	8	ap_none	b_1_0	pointer
b_1_1	in	8	ap_none	b_1_1	pointer
b_1_2	in	8	ap_none	b_1_2	pointer
b_2_0	in	8	ap_none	b_2_0	pointer
b_2_1	in	8	ap_none	b_2_1	pointer
b_2_2	in	8	ap_none	b_2_2	pointer
res_0_address0	out	2	ap_memory	res_0	array
res_0_ce0	out	1	ap_memory	res_0	array
res_0_we0	out	1	ap_memory	res_0	array
res_0_d0	out	16	ap_memory	res_0	array
res_1_address0	out	2	ap_memory	res_1	array
res_1_ce0	out	1	ap_memory	res_1	array
res_1_we0	out	1	ap_memory	res_1	array
res_1_d0	out	16	ap_memory	res_1	array
res_2_address0	out	2	ap_memory	res_2	array
res_2_ce0	out	1	ap_memory	res_2	array
res_2_we0	out	1	ap_memory	res_2	array
res_2_d0	out	16	ap_memory	res_2	array

Function level

```
// Iterate over the rows of the A matrix
Row: for(int i = 0; i < MAT_A_ROWS; i++) {
    // Iterate over the columns of the B matrix
    Col: for(int j = 0; j < MAT_B_COLS; j++) {
        res[i][j] = 0;
        // Do the inner product of a row of A and col of B
        Product: for(int k = 0; k < MAT_B_ROWS; k++) {
            res[i][j] += a[i][k] * b[k][j];
```

Vivado_hls

Outline | Directive

- matrixmul
  - % HLS PIPELINE
  - a
  - % HLS ARRAY_RESHAPE reshape variable=a complete dim=2
  - b
  - % HLS ARRAY_RESHAPE reshape variable=b complete dim=1
  - res
  - Row
    - Col

Timing (ns)

Clock		solution4	solution6
ap_clk	Target	13.33	13.33
	Estimated	7.566	7.566

Latency (clock cycles)

		solution4	solution6
Latency	min	11	5
	max	11	5
Interval	min	11	5
	max	11	5

Utilization Estimates

	solution4	solution6
BRAM_18K	0	0
DSP48E	2	18
FF	18	343
LUT	187	565
URAM	0	0

Vitis_hls

- Latency = 30ns，II=1, 27 multipliers
- 27 DSP ~7x ，FF 4x LUT 2x

Timing

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00 ns	2.936 ns	1.35 ns

Latency

Summary

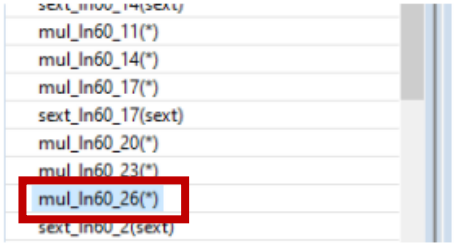
Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
6	6	30.000 ns	30.000 ns	1	1	yes

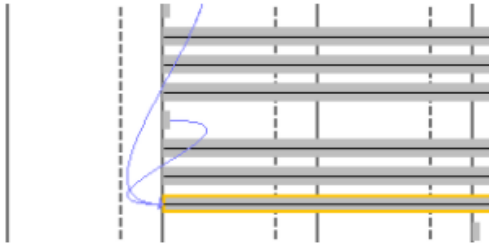
Detail

- Instance
- Loop

Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	18	-	-	-
Expression	-	-	0	2	-
FIFO	-	-	-	-	-
Instance	-	9	0	0	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	-	-
Register	-	-	775	192	-
Total	0	27	775	194	0
Available	280	220	106400	53200	0
Utilization (%)	0	12	~0	~0	0





Appendix

Reshape v.s. Partition

Reshape v.s. Partition

- Pipeline @ col loop, 6x6 matmul
- Little different in cycles

- However, # of Multiplier diffs
  - Partition → 6 Multiplier
  - Reshape → 1 Multiplier

Performance Estimates

Timing

Clock		solution11_4_partition	solution4
ap_clk	Target	10.00 ns	10.00 ns
	Estimated	6.499 ns	6.492 ns

Latency

		solution11_4_partition	solution4
Latency (cycles)	min	42	39
	max	42	39
Latency (absolute)	min	0.420 us	0.390 us
	max	0.420 us	0.390 us
Interval (cycles)	min	43	40
	max	43	40

Utilization Estimates

	solution11_4_partition	solution4
BRAM_18K	0	0
DSP	4	0
FF	258	49
LUT	353	204
URAM	0	0

- Reshape can possibly have better performance under some cases.

Streaming

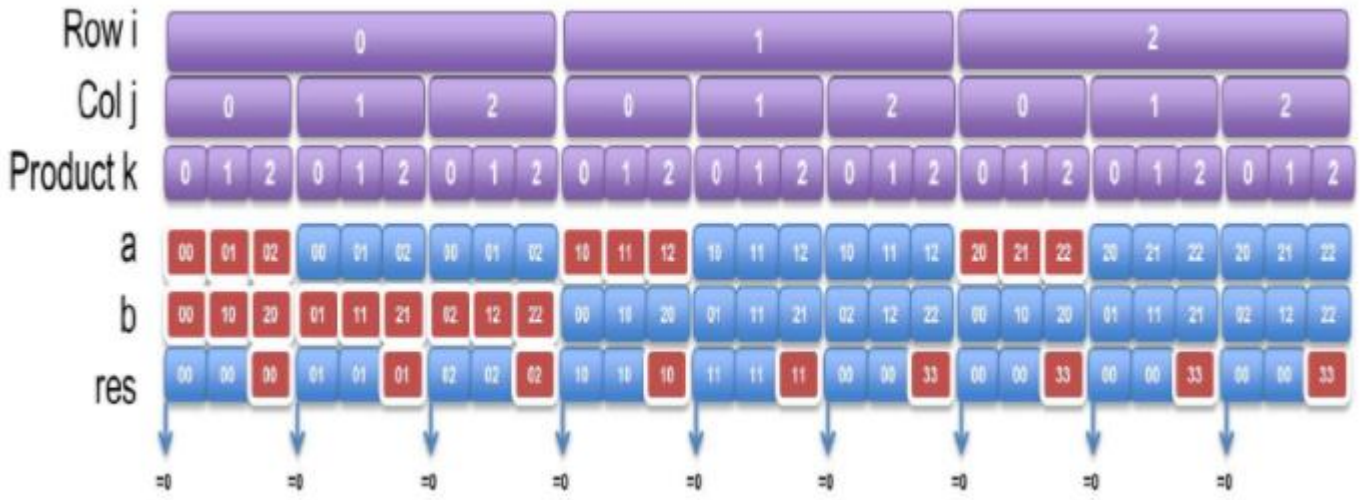
Streaming

```
// Iterate over the rows of the A matrix
Row: for(int i = 0; i < MAT_A_ROWS; i++) {
  // Iterate over the columns of the B matrix
  Col: for(int j = 0; j < MAT_B_COLS; j++) {
    res[i][j] = 0;
    // Do the inner product of a row of A and col of B
    Product: for(int k = 0; k < MAT_B_ROWS; k++) {
      res[i][j] += a[i][k] * b[k][j];
```

- Non-sequential data
  - Write res[0][0] = 0
  - Write res[0][0], k=1
  - Write res[0][0], k=2
  - Write res[0][0], k=3

```
Console  Errors  Warnings  DRCs
Vivado HLS Console
INFO: [HLS 200-10] Opening project 'C:/Vivado_HLS_Tutorial/Design_Optimization/lab1/matrixmul_prj'.
INFO: [HLS 200-10] Adding design file 'matrixmul.cpp' to the project
INFO: [HLS 200-10] Adding test bench file 'matrixmul_test.cpp' to the project
INFO: [HLS 200-10] Opening solution 'C:/Vivado_HLS_Tutorial/Design_Optimization/lab1/matrixmul_prj/solution5'
INFO: [SYN 201-201] Setting up clock 'default' with a period of 13.333ns.
INFO: [HLS 200-10] Setting target device to 'xcvu9p-flgb2104-1-e'
INFO: [SCHED 204-61] Option 'relax_ii_for_timing' is enabled, will increase II to preserve clock frequency co
INFO: [HLS 200-10] Analyzing design file 'matrixmul.cpp' ...
INFO: [HLS 200-111] Finished Linking Time (s): cpu = 00:00:01 ; elapsed = 00:00:10 . Memory (MB): peak = 105.
INFO: [HLS 200-111] Finished Checking Pragmas Time (s): cpu = 00:00:01 ; elapsed = 00:00:10 . Memory (MB): pe
INFO: [HLS 200-10] Starting code transformations ...
INFO: [HLS 200-111] Finished Standard Transforms Time (s): cpu = 00:00:01 ; elapsed = 00:00:10 . Memory (MB):
INFO: [HLS 200-10] Checking synthesizability ...
ERROR: [SYNCHK 200-91] Port 'res' (matrixmul.cpp:48) of function 'matrixmul' cannot be set to a FIFO
ERROR: [SYNCHK 200-91] as it has both write (matrixmul.cpp:60:13) and read (matrixmul.cpp:60:13) operations.
```

- Blue indicates repeated R/W
- Red indicates the first R/W



Code reorder + bug fix

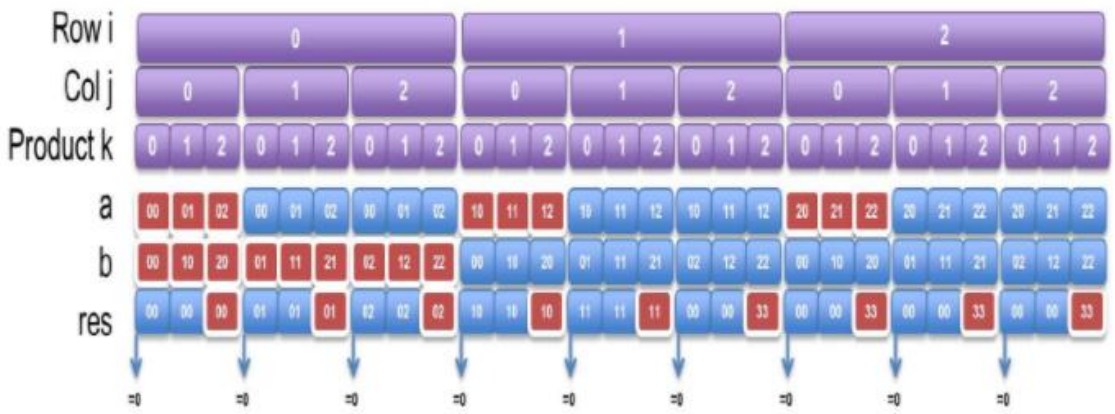
```
void matrixmul(
      mat_a_t a[MAT_A_ROWS][MAT_A_COLS],
      mat_b_t b[MAT_B_ROWS][MAT_B_COLS],
      result_t res[MAT_A_ROWS][MAT_B_COLS])
{
  mat_a_t a_row[MAT_A_ROWS];
  mat_b_t b_copy[MAT_B_ROWS][MAT_B_COLS];
  int tmp = 0;

  // Iterate over the rowa of the A matrix
  Row: for(int i = 0; i < MAT_A_ROWS; i++) {
    // Iterate over the columns of the B matrix
    Col: for(int j = 0; j < MAT_B_COLS; j++) {

      // Do the inner product of a row of A and col of B
      tmp=0;
      // Cache each row (so it's only read once per function)
      if (j == 0)
        Cache_Row: for(int k = 0; k < MAT_A_COLS; k++)
          a_row[k] = a[i][k];

       // Cache all cols (so they are only read once per function)
      if (i == 0)
            Cache_Col: for(int k = 0; k < MAT_B_ROWS; k++)
               b_copy[k][j] = b[k][j];

      Product: for(int k = 0; k < MAT_B_ROWS; k++) {
        tmp += a_row[k] * b_copy[k][j];
      }
      res[i][j] = tmp;
    }
  }
}
```



K < MAT_A_COLS

The original ug871 code is

K < MAT_A_ROWS